

Automatic Temperature Setpoint Tuning of a Thermoforming Machine using Fuzzy Terminal Iterative Learning Control

Mathieu Beauchemin-Turcotte*, Guy Gauthier[†]
and Robert Sabourin*

March 30, 2017

Abstract

This paper presents a new way to design a Fuzzy Terminal Iterative Learning Control (TILC) to control the heater temperature setpoints of a thermoforming machine. This fuzzy TILC is based on the inverse of a fuzzy model of this machine, and is built from experimental (or simulation) data with kriging interpolation. The Fuzzy Inference System usually used for a fuzzy model is the zero order Takagi Sugeno Kwan system (constant consequents). In this paper, the 1st order Takagi Sugeno Kwan system is used, with the fuzzy model rules expressed using matrices. This makes the inversion of the fuzzy model much easier than the inversion of the fuzzy model based on the TSK of order 0. Based on simulation results, the proposed fuzzy TILC seems able to give a very good initial guess as to the heater temperature setpoints, making it possible to have almost no wastage of plastic sheets. Simulation results show the effectiveness of the fuzzy TILC compared to a crisp TILC, even though the fuzzy controller is based on a fuzzy model built from noisy data.

Keywords

Fuzzy internal model control, fuzzy modeling, kriging interpolation, terminal iterative learning control.

*Affiliated with Department of automated engineering, École de technologie supérieure, Montréal, Canada.

[†]Corresponding author: guy.gauthier@etsmtl.ca

1 Introduction

The thermoforming process plays an important role in the plastics industry. In North America in 2003, in excess of \$10 billion worth of thermoformed parts were produced [1]. Examples of the parts fabricated using this process include car bumpers, baths, and packaging products, among numerous others.

The plastic sheets used in the thermoforming process undergo three phases [2, 3, 4]. The reheat phase, where the plastic sheet is heated to a sufficiently high temperature to render it malleable; the molding phase; and the cooling phase. The malleable plastic sheet is draped over a mold to give it the desired shape, and then cooled in the mold until it becomes rigid again. Finally, in the trimming phase, excess plastic is removed to obtain a usable part.

In this paper, we address only issues arising in the reheat phase of the thermoforming process. This phase is critical, since the molding will not be accurate if the plastic sheet is not heated correctly [3, 5], resulting in rejection of the inaccurately molded part by the quality control department.

To heat the plastic sheet correctly, the heater temperature setpoints have to be appropriately adjusted. This adjustment is performed by the operator of the machine using a trial-and-error approach. This approach results in plastic sheets being wasted, and occurs mostly at the start of a new production lot. It also occurs when the temperature in the plant changes significantly after the adjustment is made. In North America, plastic sheet wastage during the manufacture of a component can represent up to 10 to 15% of each production lot. For complicated parts, this percentage can rise to 20% [3].

An experienced operator can successfully reduce these losses. However, plastics processing companies are struggling to find qualified and experienced people to operate their thermoforming machines. Their fear now is that a large number of these employees will reach retirement age over the next few years.

Very little work has been published regarding the control of sheet surface temperature [6]. Such a control system would improve the quality of the parts produced, reduce scrap, and allow for temperature zoning. Plastic sheet temperature is controlled by adjusting the heater temperature of the thermoforming machine, and this control must be such that the plastic sheet temperature converges to a desired temperature profile. This issue belongs to a class of problems known as "inverse heating problems" (IHP) [7].

Existing TILC algorithms are highly dependent on initial guesswork with regard to temperature setpoints. When the initial guess is near the temperature setpoint required to heat a plastic sheet correctly, all is well. But, when it is not, a number of cycles are required to converge to the right value, causing wastage of plastic sheets. By increasing the controller gain, the convergence speed can be improved, but at the cost of less robustness. Furthermore, with those high gains, the system can be subject to overshoot. For a thermoforming oven, this is a serious problem, since there is a risk that a very malleable, overheated plastic sheet will fall on the bottom heaters, forcing the process to be stopped in order to clean them.

A suitable approach that makes this initial guess reliably would be a major

improvement in the control of this kind of process, where money needs to be saved by reducing wastage. It would result in the right heater temperature setpoints being determined at the start of the process, and reduce the controller gain perturbations, improving robustness.

The selected control approach is a TILC based on an inverse fuzzy model. This nonlinear control approach permits a relatively good initial guess of heater temperature setpoints from a desired surface temperature profile of the plastic sheet. The design approach presented here provides the opportunity to use data from a reliable nonlinear model, or from the process itself. This fuzzy approach has never been used for a TILC. The main contribution of this paper is the use of the 1st order Takagi Sugeno Kwan Fuzzy Inference System, along with its rule consequent, expressed in matrix form and built from kriging interpolation.

In this paper, we make a number of assumptions: 1) That the ambient temperature is constant (this assumption is relaxed in the simulations results); 2) That the initial temperature of the plastic sheet is constant; 3) That the plastic sheet is homogeneous and not subject to sag; 4) That the PID loops, in which the setpoints interact with the heater [4], have reached their steady state when the plastic sheet is heated; 5) That the number of inputs is equal to the number of outputs; and 6) That the minimal and maximal output values are located at opposite corners of the input space. This last assumption fits well with the way the thermoforming oven functions, since the plastic sheet surface temperature is minimal when all the heaters are at minimal temperature, and maximal when all the heaters are at maximal temperature.

The paper is structured as follows. Related works are covered in section 2. A system overview is provided in section 3. Section 4 explains how the fuzzy model of a nonlinear process is obtained. Section 5 shows the steps to follow to invert the fuzzy model, giving the inverse fuzzy model component of the TILC (Figure 3). Section 6 concerns the fuzzy filter used in the fuzzy TILC that adjusts the heater temperature setpoints of the thermoforming oven. Section 7 presents the case of a fuzzy TILC applied to a six heater, six sensor thermoforming oven model, and simulation results demonstrate how the fuzzy TILC behaves. Finally, we provide our conclusions in section 8.

2 Related works

2.1 Introducing Terminal Iterative Learning Control

Analysis of this temperature control problem gave us the idea of using a cycle-to-cycle approach to automatically tune the heater temperature setpoints, and led to considering a Terminal Iterative Learning Control (TILC). This approach was first introduced in [8] and then in a Ph.D. thesis [9], and is derived from Iterative Learning Control (ILC) [9, 10]. It has been used for the cycle-to-cycle control of a Rapid Thermal Process and Chemical Vapor Deposition (RPTCVD) [8, 9, 11]. The main difference between ILC and TILC is that ILC uses measurements sampled along the entire cycle, while TILC only uses measurements sampled at

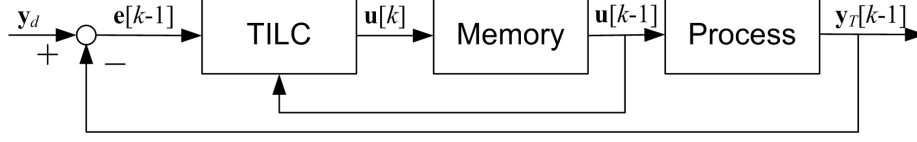


Figure 1: Block diagram of a TILC controlled system

the end of the cycle.

The TILC is based on certain assumptions [9, 10, 12]:

- That the cycle length $T \in \mathbb{R}$ has a fixed duration;
- That the input vector $\mathbf{u}[k] \in \mathbb{R}^m$ is maintained constant during the entire cycle k ($k \in \mathbb{N}$);
- That the process parameters can vary in the time domain (t) but are invariant in the cycle domain (k);
- That the process start always with the same initial state vector $\mathbf{x}_0 \in \mathbb{R}^n$, so $\mathbf{x}_0[k] = \mathbf{x}_0 \quad \forall k$;
- That the desired output vector $\mathbf{y}_d \in \mathbb{R}^p$ must be feasible, so an input vector $\mathbf{u}[k] \in \mathbb{R}^m$ must exist that made the terminal output $\mathbf{y}_T[k] \in \mathbb{R}^p$ equal to \mathbf{y}_d ; and, finally,
- That this input vector $\mathbf{u}[k]$ must be unique.

These assumptions fit relatively well with the thermoforming rehear phase. To use the TILC approach on a thermoforming machine, temperature sensors are installed to measure the surface temperature of the plastic sheet at the end of the heating cycle [4, 6, 12]. TILC iteratively adjusts the heater temperature setpoints, so that the sheet surface temperature converges to the desired level at the end of the heating cycle [12].

The block diagram of a TILC algorithm is shown in Figure 1. The input vector $\mathbf{u}[k]$, applied to the process during the entire cycle k , is a function of the terminal output vector $\mathbf{y}_T[k-1]$ measured at the end of the previous cycle ($k-1$), the corresponding input vector $\mathbf{u}[k-1]$ and the desired terminal output vector \mathbf{y}_d . The simplest TILC algorithm (called a 1st order TILC) is expressed as follows:

$$\mathbf{u}[k] = \mathbf{u}[k-1] + \mathbf{K}(\mathbf{y}_d - \mathbf{y}_T[k-1]) \quad (1)$$

where $\mathbf{K} \in \mathbb{R}^{m \times p}$ is the matrix containing the TILC gains.

2.2 Other related control approaches

This problem has been addressed in semiconductor industry processes like the Rapid Thermal Process and Chemical Vapor Deposition (RPTCVD), by the use of run-to-run (R2R) control. Like TILC, the R2R algorithm uses only

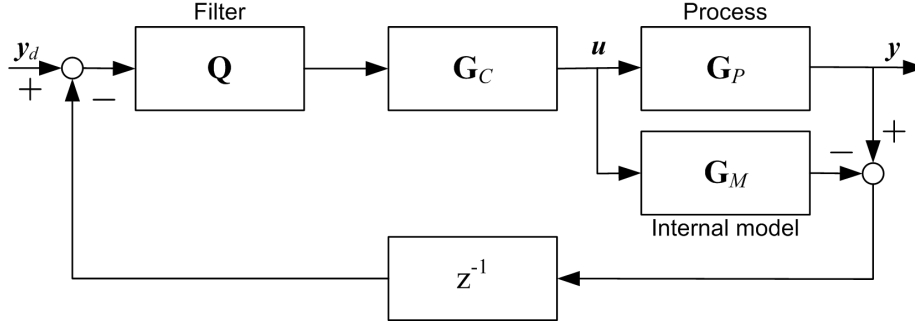


Figure 2: Internal model control structure (non-fuzzy)

measurements performed at the end of the cycle or later [13, 14, 15, 16]. The R2R approach is composed of various control algorithms [14], the one most often used being the Exponential Weighting Moving Average (EWMA) [13, 14]. R2R control is an approach derived from statistical process control.

Papers [7, 17] cover control of the temperature profile of a plastic material in roll-fed thermoforming. This control approach is based on a computational finite element algorithm, which is mainly based on the geometric relationship between the heating elements and the plastic material. These relationships are used to compute the heater temperature setpoints. In [7], this inverse problem is considered to be an ill-posed problem, since two sets of heater temperature setpoints can give relatively similar surface temperature profiles. In [17], the sensitivity of the algorithm proposed in [7] to perturbations is studied.

In a recent paper [18], the plastic sheet temperature is controlled with a two-dimensional fast Fourier transform (FFT) of the sheet temperature profile. This approach is also explained in a book chapter [19]. These authors have also proposed an alternative approach, based on the conjugate gradient method [5].

3 System overview

The fuzzy TILC (f-TILC) proposed in this paper is based on a TILC using a fuzzy nonlinear internal model control (FNIMC). The non fuzzy TILC with Internal Model Control (IMC) proposed in the chapter 3 of [12] has the structure shown in Figure 2 (where \mathbf{G}_P is the process, \mathbf{G}_M is the process model, \mathbf{G}_C is the controller and \mathbf{Q} is the filter).

The block diagram of the proposed fuzzy TILC with IMC is shown in Figure 3. This block diagram looks similar to the FNIMC block diagram shown in Figure 3 of [20], except for the z^{-1} block in the feedback.

In the block diagram of the proposed fuzzy TILC with IMC, the filter is also a fuzzy function that permits the TILC to exhibit a behavior, such that the convergence of the output vector $\mathbf{y}_T[k]$ to \mathbf{y}_d is relatively monotonic. It is used to evaluate the corrected setpoint for the next cycle $\mathbf{sp}[k]$, which in turn is used

to evaluate the input $\mathbf{u}[k]$. The initial value $\mathbf{sp}[0]$ is set equal to the desired value \mathbf{y}_d .

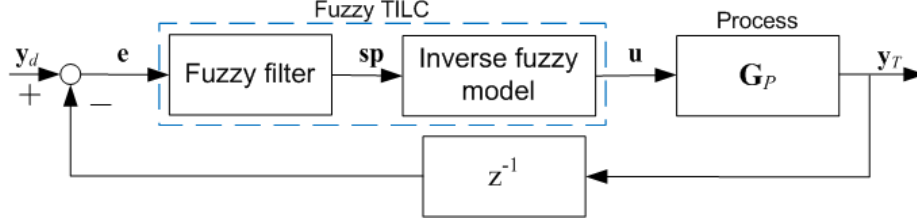


Figure 3: Proposed fuzzy TILC with FNIMC structure

The fuzzy internal model control approach has been covered by a few researchers who have considered the Takagi-Sugeno fuzzy model with constant consequents [20, 21, 22, 23, 24].

In [21], the modeled nonlinear system with fuzzy logic is a multiple input single output (MISO) system that is converted into a single input single output (SISO) system before being inverted. This inverted fuzzy model is used in a Fuzzy Nonlinear Internal Model Control (FNIMC).

This approach was extended in [20] to a multiple input multiple output (MIMO) process. These authors consider a class of MIMO processes that can be decomposed into MISO subsystems, each modeled using fuzzy logic as in [21]. The Fuzzy Inference System (FIS) used for the fuzzy model is the Takagi-Sugeno-Kwan (TSK) FIS of order 0.

In [23], the inversion of a fuzzy model is presented. Some of the mathematical notations used in this paper were inspired to some extent by [23]. In [24], a nonlinear process is decomposed using a fuzzy decoupling controller for a nuclear power plant.

Since the thermoforming process is a nonlinear MIMO system that can be decomposed into MISO subsystems, the approach presented in [20] is applicable. In this paper, the FIS used is the TSK of order 1, and the output of each rule is a linear polynomial. At the defuzzification stage, the combination of those linear rules with membership functions generates nonlinear outputs. The approach used for the SISO fuzzy model inversion is extended to MIMO fuzzy model inversion by expressing the rules in matrix form. The rules of the inverse of the model are calculated using a matrix calculation. A 1st order TSK FIS for the fuzzy model provides an easier way to invert the fuzzy model than the TSK FIS of order 0, especially for large MIMO systems.

The design of the proposed fuzzy TILC is based on the inverse of the fuzzy model of the process (Figure 3). To be able to create this design, we first need to obtain the fuzzy model of the nonlinear process, which we do from experiments performed on the process, by applying inputs and measuring the corresponding outputs. The fuzzy model of the process is based on the Takagi Sugeno Kwan (TSK) fuzzy inference system (FIS) of order 1.

The fuzzy model is built in five steps. The first step is to define the universe

of discourse of each input to the process, which corresponds to the variation between the minimum and maximum value of each input. The second step is to divide each defined universe of discourse into a number of fuzzy sets. The third step is to define the input vectors (also called input tuples) that need to be applied to the process at the experimental stage, in order to obtain the rules of the fuzzy model. In the fourth step, which is the experimental stage, a database is built from the input vectors applied to the process and the corresponding output vectors measured in the experiments. The last step is to evaluate the rules of the fuzzy model from the database of input and output vectors. This is done using an approach not used in fuzzy control: kriging interpolation.

4 The fuzzy modeling of nonlinear processes

The following subsections explain each of the five steps listed above. The notation used in this paper is introduced in each subsection.

4.1 Defining the universe of discourse of each input

From the knowledge available about the nonlinear process, the universe of discourse of each input is defined to make it possible to build a fuzzy model of the process. The minimal value of the j -th input $u_j \in \mathbb{R}$ is defined by $u_{min,j}$ and its maximal value by $u_{max,j}$. Then, the universe of discourse of the j -th input is $u_j \in \mathfrak{U}_j = [u_{min,j}, u_{max,j}] \subset \mathbb{R}$.

The orthogonal combination of all the universes of discourses of the m inputs leads to an m dimension input space:

$$\mathfrak{U} = \mathfrak{U}_1 \times \dots \times \mathfrak{U}_m \in \mathbb{R}^m \quad (2)$$

so the input vector $\mathbf{u} = [u_1, \dots, u_m]^T \in \mathfrak{U}$.

The universes of discourse are divided into a number of fuzzy sets.

4.2 Dividing the universe of discourse of each input into fuzzy sets

The universe of each input is covered by a certain number of fuzzy sets (Figure 4), each of which is identified by a linguistic term. The i -th fuzzy set of the j -th input is identified by the linguistic term $A_{i,j}$. The number of fuzzy sets defined for the j -th input is identified by $N_j \in \mathbb{N}$. The number of fuzzy sets has an effect on the accuracy of the fuzzy model.

For the proposed fuzzy model, the membership functions of each fuzzy set are triangular (see Figure 4). When $i \in \{2, \dots, N_j - 1\}$, the membership functions are defined as follows:

$$\mu_{A_{i,j}}(u_j) = \begin{cases} \mathbf{n}_1/\mathbf{m}_1 & a_{i-1,j} \leq u_j \leq a_{i,j} \\ \mathbf{n}_2/\mathbf{m}_2 & a_{i,j} < u_j \leq a_{i+1,j} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

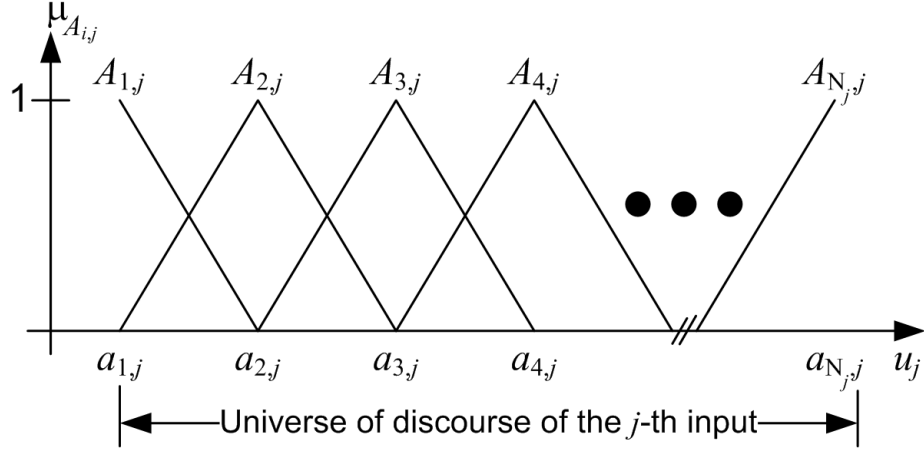


Figure 4: Membership functions of the j -th input fuzzy sets

where $\mathbf{n}_1 = u_j - a_{i-1,j}$, $\mathbf{m}_1 = a_{i,j} - a_{i-1,j}$, $\mathbf{n}_2 = a_{i+1,j} - u_j$, $\mathbf{m}_2 = a_{i+1,j} - a_{i,j}$ and $a_{i,j}$ is the abscissa of the peak of the $A_{i,j}$ fuzzy set (Figure 4).

When $i = 1$, we have the membership level of the left-most fuzzy set ($A_{1,j}$):

$$\mu_{A_{1,j}}(u_j) = \begin{cases} 1 & u_j \leq u_{min,j} \\ \mathbf{n}_3/\mathbf{m}_3 & u_{min,j} < u_j \leq a_{2,j} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $\mathbf{n}_3 = u_j - u_{min,j}$ and $\mathbf{m}_3 = a_{2,j} - u_{min,j}$

When $i = N_j$, we have the membership level of the right-most fuzzy set ($A_{N_j,j}$):

$$\mu_{A_{N_j,j}}(u_j) = \begin{cases} 1 & u_j \geq u_{max,j} \\ \mathbf{n}_4/\mathbf{m}_4 & a_{N_j-1,j} \leq u_j < u_{max,j} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $\mathbf{n}_4 = u_{max,j} - u_j$ and $\mathbf{m}_4 = u_{max,j} - a_{N_j-1,j}$.

For the proposed fuzzy model, the partitioning of the universe of discourse needs to be strict, as in [21][23]. So, for two adjacent fuzzy sets $A_{i,j}$ and $A_{i+1,j}$, the sum of their membership functions is:

$$\mu_{A_{i,j}}(u_j) + \mu_{A_{i+1,j}}(u_j) = 1 \quad (6)$$

where $a_{i,j} \leq u_j \leq a_{i+1,j}$, for all $i \in \{1, \dots, N_j - 1\}$ and all the m inputs. Therefore, for a given input u_j with a value within the universe of discourse, there are only two fuzzy sets with a value different from zero.

All the fuzzy sets have to be implemented to obtain the fuzzification component of the fuzzy model. Since the j -th input has N_j fuzzy sets, the total number of fuzzy sets in the fuzzy model is $N_{FS} = N_1 + \dots + N_m$.

4.3 Defining the required input vectors necessary for experimentation on the process

In fuzzy logic, each rule is associated with a combination of the fuzzy sets of the inputs. For example, rule $\mathfrak{R}_{i_1, \dots, i_m, k}$ is defined as follows:

$$\mathfrak{R}_{i_1, \dots, i_m, k} : \text{If } u_1 \text{ is } A_{i_1, 1} \text{ and } \dots \text{ and } u_m \text{ is } A_{i_m, m}, \text{ then } y_k = r_{i_1, \dots, i_m, k} \quad (7)$$

In (7), k is the output number, and $i_j \in \{1, \dots, N_j\}$ is used to identify the fuzzy set number considered in the rule for the j -th input. The consequent of rule $\mathfrak{R}_{i_1, \dots, i_m, k}$ is expressed by $r_{i_1, \dots, i_m, k}$. The k -th output y_k is obtained from the rules consequent.

In this paper, we propose to create the fuzzy model rules from a database built from experiments performed on the process. The input vectors (also called input tuples) used in the experiments are defined from the fuzzy sets defined in the previous step. In the input tuple, the j -th input u_j is such that $u_j \in \{u_{min, j}, b_{1, j}, \dots, b_{N_j-1, j}, u_{max, j}\}$, where the $b_{i, j}$ abscissa are:

$$b_{i, j} = (a_{i, j} + a_{i+1, j})/2, \quad i \in \{1, \dots, N_j - 1\} \quad (8)$$

and this is done for all the m inputs ($j \in \{1, \dots, m\}$).

The input tuples used to build the database are identified by the vector $\theta_{\{i_1, \dots, i_m\}}$ where the indices $i_j \in \{1, \dots, N_j + 1\}$.

Since there are m inputs, the results of 2^m experiments are required for each rule. As the number of rules for the j -th input is defined as N_j , the total number of rules required for each output is $N = N_1 \times \dots \times N_m$. The total number of experiments required to obtain all N rules of the fuzzy model is $N_{experiments} = (N_1 + 1) \times \dots \times (N_m + 1)$. The results of 2^m experiments are used to obtain each rule.

4.3.1 Example

To help explain the notation, we consider a system with two inputs. Four (2^2) measurements are required to obtain rule $\mathfrak{R}_{1,1,k}$, defined as:

$$\mathfrak{R}_{1,1,k} : \text{If } u_1 \text{ is } A_{1,1} \text{ and } u_2 \text{ is } A_{1,2}, \text{ then } y_k = r_{1,1,k} \quad (9)$$

The input tuples applied to the system for those four measurements are: $\theta_{\{1,1\}} = [u_{min,1} \ u_{min,2}]^T$, $\theta_{\{1,2\}} = [u_{min,1} \ b_{1,2}]^T$, $\theta_{\{2,1\}} = [b_{1,1} \ u_{min,2}]^T$ and $\theta_{\{2,2\}} = [b_{1,1} \ b_{1,2}]^T$. Then, the four experiments are performed by setting $\mathbf{u} = \theta_{\{i,j\}}$ with $i, j \in \{1, 2\}$. ♦

4.4 Building the rules from the database

All the experiments performed on the process provide a database linking the measured output vectors $\phi_{\{i_1, \dots, i_m\}} \in \mathbb{R}^m$ with each corresponding input vector (or tuple) $\theta_{\{i_1, \dots, i_m\}} \in \mathbb{R}^m$ (with $i_j \in \{1, \dots, N_j + 1\}$). The rules are obtained from those experiments. The $\mathfrak{R}_{l_1, \dots, l_m, k}$ rule of the k -th output is obtained from

the results of 2^m experiments, where the corresponding input tuples are on the following list:

$$\{\forall \boldsymbol{\theta}_{\{i_1, \dots, i_m\}} \mid i_j \in \{l_j, l_j + 1\}\} \quad (10)$$

and the corresponding k -th output measurements are also on this list:

$$\{\forall \phi_{\{i_1, \dots, i_m\}, k} \mid i_j \in \{l_j, l_j + 1\}\} \quad (11)$$

The cardinality of these two lists is 2^m .

The two lists, (10) and (11), are used to find the consequent $r_{l_1, \dots, l_m, k}$ of the rule $\mathfrak{R}_{l_1, \dots, l_m, k}$ by using kriging interpolation. The contents of the two lists are arranged in an $m \times 2^m$ matrix defined as:

$$\boldsymbol{\Theta}_{\{l_1, \dots, l_m\}} = \begin{bmatrix} \boldsymbol{\theta}_{\{l_1, \dots, l_m\}} & \cdots & \boldsymbol{\theta}_{\{l_1+1, \dots, l_m+1\}} \end{bmatrix}^T \quad (12)$$

and the corresponding k -th output vector (of size 2^m):

$$\mathbf{Y}_{\{l_1, \dots, l_m\}, k} = \begin{bmatrix} \phi_{\{l_1, \dots, l_m\}, k} & \cdots & \phi_{\{l_1+1, \dots, l_m+1\}, k} \end{bmatrix}^T \quad (13)$$

4.4.1 Example (con't)

Continuing the example with two inputs, the matrices used to find the rule $\mathfrak{R}_{1,1,k}$ are:

$$\boldsymbol{\Theta}_{\{1,1\}} = \begin{bmatrix} \boldsymbol{\theta}_{\{1,1\}} & \boldsymbol{\theta}_{\{1,2\}} & \boldsymbol{\theta}_{\{2,1\}} & \boldsymbol{\theta}_{\{2,2\}} \end{bmatrix}^T \quad (14)$$

and :

$$\mathbf{Y}_{\{1,1\}, k} = \begin{bmatrix} \phi_{\{1,1\}, k} & \phi_{\{1,2\}, k} & \phi_{\{2,1\}, k} & \phi_{\{2,2\}, k} \end{bmatrix}^T \quad (15)$$

◆

Kriging [25] is the approach used to perform the interpolation to obtain the rules of the fuzzy model [26]. The equation used by the kriging interpolation to obtain the coefficients of the rule consequent $r_{l_1, \dots, l_m, k}$ is the following linear system:

$$\begin{bmatrix} \mathbf{I}_{2^m} & \mathbf{1}_{2^m}^T & \boldsymbol{\Theta}_{\{l_1, \dots, l_m\}} \\ \mathbf{1}_{2^m} & 0 & \mathbf{0}_m \\ \boldsymbol{\Theta}_{\{l_1, \dots, l_m\}}^T & \mathbf{0}_m^T & \mathbf{0}_{m \times m} \end{bmatrix} \times \begin{bmatrix} \mathbf{B}_{\{l_1, \dots, l_m\}, k} \\ c_{\{l_1, \dots, l_m\}, 0, k} \\ \mathbf{A}_{\{l_1, \dots, l_m\}, k} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{\{l_1, \dots, l_m\}, k} \\ 0 \\ \mathbf{0}_m \end{bmatrix} \quad (16)$$

In (16), $\mathbf{I}_{2^m} \in \mathbb{R}^{2^m \times 2^m}$ represents the identity matrix, $\mathbf{0}_{m \times m} \in \mathbb{R}^{m \times m}$ is a square matrix filled with zeros, $\mathbf{1}_{2^m} \in \mathbb{R}^{2^m}$ is a vector filled with ones and $\mathbf{0}_m \in \mathbb{R}^m$ is a vector filled with zeros. The vector $\mathbf{A}_{\{l_1, \dots, l_m\}, k} \in \mathbb{R}^m$ contains the list of the coefficients of the rule consequent $r_{l_1, \dots, l_m, k}$:

$$\mathbf{A}_{\{l_1, \dots, l_m\}, k} = \begin{bmatrix} c_{\{l_1, \dots, l_m\}, 1, k} & \cdots & c_{\{l_1, \dots, l_m\}, m, k} \end{bmatrix}^T \quad (17)$$

The constant coefficient of the rule $r_{l_1, \dots, l_m, k}$ is the variable $c_{\{l_1, \dots, l_m\}, 0, k}$. From the coefficients contained in $\mathbf{A}_{\{l_1, \dots, l_m\}, k}$ and $c_{\{l_1, \dots, l_m\}, 0, k}$, the rule consequent $r_{l_1, \dots, l_m, k}$ of a TSK FIS of order 1 is defined by this affine function:

$$r_{l_1, \dots, l_m, k} = c_{\{l_1, \dots, l_m\}, 0, k} + \sum_{j=1}^m (c_{\{l_1, \dots, l_m\}, j, k} \times u_j) \quad (18)$$

The vector $\mathbf{B}_{\{l_1, \dots, l_m\}, k} \in \mathbb{R}^{2^m}$ contains the information about the interpolation accuracy [25]. When this vector is filled with zeros, the kriging interpolation gives the exact model for the subspace corresponding to the rule $\mathfrak{R}_{i_1, \dots, i_m, k}$, based on the 2^m measurements.

The coefficients of the rule $\mathfrak{R}_{i_1, \dots, i_m, k}$ and the vector $\mathbf{B}_{\{l_1, \dots, l_m\}, k}$ are obtained by solving the linear system (16). This interpolation process is repeated $N \times m$ times, since there are N rules consequent for each of the m outputs.

From the interpolated rule consequent $r_{i_1, \dots, i_m, k}$, we are able to build vectors and matrices. In fact, the rule consequents are grouped into vectors $\mathbf{R}_{\{l_1, \dots, l_m\}} \in \mathbb{R}^m$ as follows:

$$\mathbf{R}_{\{l_1, \dots, l_m\}} = [r_{l_1, \dots, l_m, 1} \quad \dots \quad r_{l_1, \dots, l_m, m}]^T \quad (19)$$

and each of these vectors is defined as follows:

$$\mathbf{R}_{\{l_1, \dots, l_m\}} = \mathbf{C}_{\{l_1, \dots, l_m\}} + \mathbf{D}_{\{l_1, \dots, l_m\}}^T \mathbf{u} \quad (20)$$

In (20), $\mathbf{C}_{\{l_1, \dots, l_m\}} \in \mathbb{R}^m$ is a vector containing the $c_{\{l_1, \dots, l_m\}, 0, k}$ terms of the rules:

$$\mathbf{C}_{\{l_1, \dots, l_m\}} = [c_{\{l_1, \dots, l_m\}, 0, 1} \quad \dots \quad c_{\{l_1, \dots, l_m\}, 0, m}]^T \quad (21)$$

and $\mathbf{D}_{\{l_1, \dots, l_m\}} \in \mathbb{R}^{m \times m}$ is a matrix containing the $c_{\{l_1, \dots, l_m\}, j, k}$ terms (with $j \neq 0$) of the rules:

$$\mathbf{D}_{\{l_1, \dots, l_m\}} = \begin{bmatrix} c_{\{l_1, \dots, l_m\}, 1, 1} & \dots & c_{\{l_1, \dots, l_m\}, m, 1} \\ \vdots & \ddots & \vdots \\ c_{\{l_1, \dots, l_m\}, 1, m} & \dots & c_{\{l_1, \dots, l_m\}, m, m} \end{bmatrix} \quad (22)$$

All the inputs of the fuzzy model are combined into a vector $\mathbf{u} = [u_1, \dots, u_m]^T \in \mathbb{R}^m$.

As the kriging interpolation is linear, it is possible to obtain the matrix \mathbf{D} and the vector \mathbf{C} defined in (22) and (21) directly, since it is possible to rewrite (16) as follows:

$$\begin{bmatrix} \mathbf{I}_{2^m} & \mathbf{1}_{2^m}^T & \boldsymbol{\Theta}_{\{l_1, \dots, l_m\}} \\ \mathbf{1}_{2^m} & 0 & \mathbf{0}_m \\ \boldsymbol{\Theta}_{\{l_1, \dots, l_m\}}^T & \mathbf{0}_m^T & \mathbf{0}_{m \times m} \end{bmatrix} \times \begin{bmatrix} \mathbf{B}_{\{l_1, \dots, l_m\}} \\ \mathbf{C}_{\{l_1, \dots, l_m\}}^T \\ \mathbf{D}_{\{l_1, \dots, l_m\}}^T \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{\{l_1, \dots, l_m\}} \\ 0 \\ \mathbf{0}_m \end{bmatrix} \quad (23)$$

where the matrices $\mathbf{B}_{\{l_1, \dots, l_m\}} \in \mathbb{R}^{m \times 2^m}$ and $\mathbf{Y}_{\{l_1, \dots, l_m\}} \in \mathbb{R}^{m \times 2^m}$ are respectively:

$$\mathbf{B}_{\{l_1, \dots, l_m\}} = [\mathbf{B}_{\{l_1, \dots, l_m\}, 1} \quad \dots \quad \mathbf{B}_{\{l_1, \dots, l_m\}, m}] \quad (24)$$

and

$$\mathbf{Y}_{\{l_1, \dots, l_m\}} = \begin{bmatrix} \mathbf{Y}_{\{l_1, \dots, l_m\}, 1} & \dots & \mathbf{Y}_{\{l_1, \dots, l_m\}, m} \end{bmatrix} \quad (25)$$

By solving the linear system defined in (23), the vector $\mathbf{C}_{\{l_1, \dots, l_m\}}$ and the matrix $\mathbf{D}_{\{l_1, \dots, l_m\}}$ are calculated directly.

When the rule consequent are represented vectorially, the $\mathfrak{R}_{i_1, \dots, i_m, k}$ rules, defined in (7) with $k \in \{1, \dots, m\}$, are combined into this rule:

$$\mathfrak{R}_{i_1, \dots, i_m} : \text{If } u_1 \text{ is } A_{i_1, 1} \text{ and } \dots \text{ and } u_m \text{ is } A_{i_m, m}, \text{ then } \mathbf{y} = \mathbf{R}_{\{i_1, \dots, i_m\}} \quad (26)$$

with $\mathbf{R}_{\{i_1, \dots, i_m\}}$ defined in (20) and the output vector $\mathbf{y} = [y_1 \dots y_m]^T \in \mathbb{R}^m$.

This vectorial representation of the rule consequent simplifies the representation of the FIS rules and the inversion of the rules to obtain the inverse fuzzy model.

4.5 The resulting fuzzy model

By following the steps presented in the above section, a fuzzy model based on the TSK FIS of order 1 is obtained. To implement this model, N_j fuzzy sets must be defined for the j -th input – see equations (3) to (5). This means that there are N rules to define per output, resulting in N linear equations - see (20).

In the next section, the inversion of this fuzzy model is introduced.

5 Inversion of the fuzzy model

To design the fuzzy TILC algorithm, the fuzzy model of the MIMO process has to be inverted. Two steps are required to achieve this. The first step is to define the fuzzy sets of the inputs of the inverse fuzzy model from the fuzzy sets of the inputs of the fuzzy model. If this first step is feasible, the second step is to obtain the rule consequents of the inverse fuzzy model from the rule consequent of the fuzzy model.

5.1 Preliminary assumptions

Before explaining the inversion of the fuzzy model, we must state the following two assumptions about the TSK fuzzy model that will be inverted.

Assumption III-1 That the number of inputs is equal to the number of outputs, or $p = m$, and also that input u_i is paired with output y_i in the fuzzy model. This assumption is not very restrictive, since the input output pairing of a MIMO process can be achieved using an approach like Relative Gain Array (RGA) [27]. Once this pairing has been made, the outputs are renamed, such that output y_i is paired with input u_i .

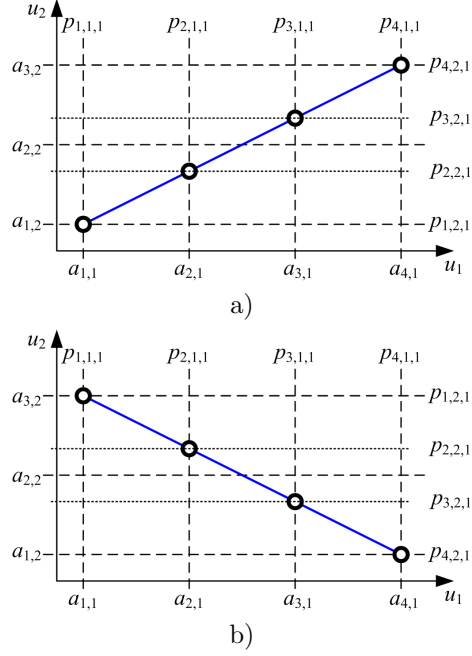


Figure 5: Diagonals along the input space \mathcal{U} of the fuzzy model for a two inputs fuzzy model

Assumption III-2 That output y_k of the fuzzy model, the minimum $y_{min,k}$ and the maximum $y_{max,k}$, are at opposite corners of the input space \mathcal{U} , and this must be the case for all outputs ($k \in \{1, \dots, m\}$).

To help in understanding the second assumption, Figure 5 shows an example for a system having two inputs. In Figure 5 a) the minimal and maximal values of the output are at the following input vectors: $\mathbf{u} = [a_{1,1} \ a_{1,2}]^T$ and $\mathbf{u} = [a_{N_1,1} \ a_{N_2,2}]^T = [a_{4,1} \ a_{3,2}]^T$. Figure 5 b) the minimal and maximal values of the output are at $\mathbf{u} = [a_{1,1} \ a_{N_2,2}]^T = [a_{1,1} \ a_{3,2}]^T$ and $\mathbf{u} = [a_{N_1,1} \ a_{1,2}]^T = [a_{4,1} \ a_{1,2}]^T$.

If neither of the cases presented in Assumption III-2 applies, the approach described in the following subsection for performing the inversion of the fuzzy model cannot be used.

5.2 Obtaining the fuzzy sets of the inverse fuzzy model

To obtain the fuzzy model of a process, fuzzy sets are defined for the m inputs of the process. In the inverse fuzzy model, the fuzzy sets are built from those fuzzy sets. Since we have assumed that the number of inputs and outputs of the process is the same, the number of fuzzy sets in the inverse fuzzy model is equal to those in the fuzzy model.

The inputs to the inverse fuzzy model are defined by the variable \tilde{y}_j , and they are grouped in the vector $\tilde{\mathbf{y}} = [\tilde{y}_1 \ \dots \ \tilde{y}_m]^T \in \mathbb{R}^m$. The linguistic term

of the i -th fuzzy set of k -input \tilde{y}_k of the inverse fuzzy model is identified by $\mathbb{A}_{i,k}$. The coordinates of the peak of each of those fuzzy sets are identified by $\mathbf{a}_{i,k}$. Each of the $\mathbf{a}_{i,k}$ coordinates is calculated from the $a_{i,j}$ coordinates from the fuzzy model in a two steps process.

To explain this process, let us consider a two input, two output fuzzy model. Figure 5 shows the universe of discourse of both the inputs and the coordinates ($a_{i,j}$) of the peaks of the fuzzy sets of both the inputs. The output y_1 of this fuzzy model is assumed to be minimal when input $u_1 = a_{1,1}$ and input $u_2 = a_{1,2}$, and maximal when $u_1 = a_{4,1}$ and $u_2 = a_{3,2}$.

Each of the peaks $\mathbf{a}_{i,1}$ of the fuzzy sets $\mathbb{A}_{i,1}$ of the input \tilde{y}_1 of the inverse fuzzy model is obtained as follows:

First, the coordinates $p_{i,j,k}$, where the fuzzy sets of the input u_1 have their peaks, are calculated. Then, for this input, the coordinates $p_{i,1,1}$ are the coordinates $a_{i,1}$ directly, and so $p_{i,1,1} = a_{i,1}$. For the other input (u_2), the coordinates to consider are identified as $p_{i,2,1}$, and are calculated by the equations below (when $j \neq k$).

For case A in Figure 5:

$$\begin{aligned} p_{i,j,k} &= \frac{(a_{N_j,j} - a_{1,j})}{(a_{N_k,k} - a_{1,k})} (a_{i,k} - a_{1,k}) + a_{1,j} \\ &= \frac{(u_{max,j} - u_{min,j})}{(u_{max,k} - u_{min,k})} (a_{i,k} - u_{min,k}) + u_{min,j} \end{aligned} \quad (27)$$

and for case B in Figure 5:

$$\begin{aligned} p_{i,j,k} &= \frac{(a_{N_j,j} - a_{1,j})}{(a_{N_k,k} - a_{1,k})} (a_{N_k,k} - a_{i,k}) + a_{1,j} \\ &= \frac{(u_{max,j} - u_{min,j})}{(u_{max,k} - u_{min,k})} (u_{max,k} - a_{i,k}) + u_{min,j} \end{aligned} \quad (28)$$

In both cases, when $j = k$ (the number of inputs and outputs is the same), the equation to use is $p_{i,k,k} = a_{i,k}$.

Finally, the coordinates $\mathbf{a}_{i,k}$ are obtained directly from the evaluation of the fuzzy model with the input vector built from those $p_{i,j,k}$ values. So, $\mathbf{a}_{i,k}$ is directly, the k -th output value y_k of the fuzzy model with the vector $[p_{i,1,k} \dots p_{i,m,k}]^T$ applied as the input vector.

In this paper, we consider the following two cases. If $\mathbf{a}_{i,k}$ are not increasing, then $\mathbf{a}_{1,k} \leq \mathbf{a}_{2,k} \leq \dots \leq \mathbf{a}_{N_k,k}$. If $\mathbf{a}_{i,k}$ are not decreasing, then $\mathbf{a}_{1,k} \geq \mathbf{a}_{2,k} \geq \dots \geq \mathbf{a}_{N_k,k}$. We check for all k (or all input output pairs), if either case applies. If neither of these two cases applies, the approach presented in this paper cannot be used. In this situation, it is not possible to design a TILC algorithm based on the inverse of the fuzzy model.

5.3 Obtaining the rules of the inverse fuzzy model

The rules of the fuzzy inverse model are defined in a similar way to the rules of the fuzzy model (7):

$$\mathcal{R}_{i_1, \dots, i_m} : \text{If } \tilde{y}_1 \text{ is } \mathbb{A}_{i_1, 1} \text{ and } \dots \text{ and } \tilde{y}_m \text{ is } \mathbb{A}_{i_m, m}, \text{ then } \tilde{u}_k = \mathfrak{r}_{i_1, \dots, i_m, k} \quad (29)$$

By grouping the rules as was done in equation (26) in the previous section, we can write:

$$\mathcal{R}_{i_1, \dots, i_m} : \text{If } \tilde{y}_1 \text{ is } \mathbb{A}_{i_1, 1} \text{ and } \dots \text{ and } \tilde{y}_m \text{ is } \mathbb{A}_{i_m, m}, \text{ then } \tilde{\mathbf{U}} = \mathbf{R}_{\{i_1, \dots, i_m\}} \quad (30)$$

where $\tilde{\mathbf{u}} = [\tilde{u}_1 \dots \tilde{u}_m]^T$ is the output vector of the inverse fuzzy model. Since the rules were expressed by the linear equations shown in (20), the inversion of the rules is calculated as follows:

$$\mathbf{R}_{\{l_1, \dots, l_m\}} = \mathbf{C}_{\{l_1, \dots, l_m\}} + \mathbf{D}_{\{l_1, \dots, l_m\}} \tilde{\mathbf{y}} \quad (31)$$

In (31), the matrix $\mathbf{D}_{\{l_1, \dots, l_m\}} \in \mathbb{R}^{m \times m}$ is calculated as follows:

$$\mathbf{D}_{\{l_1, \dots, l_m\}} = \mathbf{D}_{\{l_1, \dots, l_m\}}^{-1} \quad (32)$$

and so the vector $\mathbf{C}_{\{l_1, \dots, l_m\}} \in \mathbb{R}^m$ is the following:

$$\mathbf{C}_{\{l_1, \dots, l_m\}} = -\mathbf{D}_{\{l_1, \dots, l_m\}}^{-1} \mathbf{C}_{\{l_1, \dots, l_m\}} \quad (33)$$

As we can see, the rules for the inverse fuzzy model are easily calculated using linear algebra, i.e. by inverting the equation of the rules of the fuzzy model of the process.

Having addressed the decoupling component of the TILC algorithm, in the next section, we now cover the filter component of the TILC approach.

6 The filter component of the TILC

The IMC-based fuzzy TILC is built by combining two components: the inverse fuzzy model used for decoupling, and the filter (see Figure 3).

We have shown above how the model is obtained and inverted. The fuzzy model is an approximation of the real process, and this process can have parametric variations and external perturbations. The filter component of the TILC can compensate for model inaccuracy and perturbations [28].

6.1 Crisp version of the filter

For the IMC approach, an exponential filter is usually proposed [28]. For the i -th loop, the exponential filter is defined by:

$$\hat{Q}_i(z) = \frac{z(1 - \alpha_i)}{z - 1} \quad (34)$$

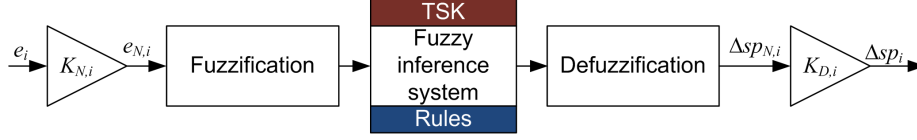


Figure 6: Block diagram of a SISO fuzzy filter

where the filter parameter $0 \leq \alpha_i < 1$ is adjusted to shape the behavior of the closed loop response. For the TILC, those α_i parameters have an effect on the convergence speed and the robustness of the closed loop system.

For a system with a MIMO IMC for m loops, the MIMO version of the exponential filter is as follows:

$$\hat{\mathbf{Q}}(z) = \frac{z}{z-1} \begin{bmatrix} 1-\alpha_1 & 0 & \dots & 0 \\ 0 & 1-\alpha_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1-\alpha_m \end{bmatrix} \quad (35)$$

6.2 Fuzzy version of the filter

The fuzzy version of the IMC filter is nonlinear, since the behavior of the filter is adjusted to the temperature error [26].

If the temperature at the surface of the plastic sheet is too high, the heater temperature setpoints have to be decreased relatively quickly. When the temperature at the surface of the plastic sheet is too low, the heater temperature setpoints have to be increased, but at a slower speed to ensure a monotonic convergence. The purpose of the nonlinear filter is to reduce the risk of high temperature setpoints, which can damage the thermoforming oven.

The proposed fuzzy filter (Figure 6) is based on a SISO TSK FIS with constant consequents [26]. The consequent of a given rule is a constant value. The i -th surface temperature error e_i is normalized to $e_{N,i}$ with a normalising gain $K_{N,i} \in \mathbb{R}$. The fuzzification component receives this normalized value and fuzzifies it with five fuzzy sets, and corresponding to those five fuzzy sets there are five rules and their corresponding consequent:

$$\mathfrak{R}_{f1} : \text{If } e_N \text{ is } A_{NB} \text{ then } \Delta u_N = 0.6 \quad (36)$$

$$\mathfrak{R}_{f2} : \text{If } e_N \text{ is } A_{NS} \text{ then } \Delta u_N = 0.25 \quad (37)$$

$$\mathfrak{R}_{f3} : \text{If } e_N \text{ is } A_{ZR} \text{ then } \Delta u_N = 0.0 \quad (38)$$

$$\mathfrak{R}_{f4} : \text{If } e_N \text{ is } A_{PS} \text{ then } \Delta u_N = -0.5 \quad (39)$$

$$\mathfrak{R}_{f5} : \text{If } e_N \text{ is } A_{PB} \text{ then } \Delta u_N = -1.0 \quad (40)$$

We can see that, as required, the temperature setpoint change $\Delta sp_{N,i}$ is greater when the sheet temperature is higher than the desired one.

The rule consequent are defuzzified, and the resulting value is defuzzified with the gain $K_{D,i} \in \mathbb{R}$ and denormalized to obtain the i -th output value Δsp_i . Then, the output of the filter is, finally: $sp_i[k+1] = sp_i[k] + \Delta sp_i[k]$. This output is sent to the fuzzy inverse model to generate the heater temperature setpoints.

The initial value $sp_i[0]$ (for cycle $k = 0$) of the fuzzy filter is equal to the desired terminal temperature $y_{d,i}$. These initial values ($\forall i \in \{1, \dots, m\}$) will be used to calculate the initial guess for the heater temperature setpoints.

7 Application to a thermoforming machine

The fuzzy TILC design approach presented in this paper is now applied to a nonlinear model of a thermoforming machine. This model, presented briefly in the Appendix, is based on the AAA thermoforming machine [12]. The simulated thermoforming machine has six inputs (heaters) and six outputs (infra-red sensors).

The six heaters are obtained by grouping the heaters bank (see Figure 7) as follows: T_1 - T_4 , T_2 - T_5 , T_3 - T_6 , B_1 - B_4 , B_2 - B_5 , and B_3 - B_6 (T identifies a top heater, and B identifies a bottom heater). Furthermore, the system input vector $\mathbf{u} \in \mathbb{R}^6$ is defined as follows:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} T_2 - T_5 \\ T_1 - T_4 \\ T_3 - T_6 \\ B_2 - B_5 \\ B_1 - B_4 \\ B_3 - B_6 \end{bmatrix} \quad (41)$$

The six infrared (IR) sensors selected are (see Figure 7): IR_{T1} , IR_{T2} , IR_{T5} , IR_{B1} , IR_{B2} , and IR_{B5} (again, T identifies a top IR sensor, B identifies a bottom IR sensor). So, the model output vector $\mathbf{y} \in \mathbb{R}^6$ is defined as follows:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} IR_{T1} \\ IR_{T2} \\ IR_{T5} \\ IR_{B1} \\ IR_{B2} \\ IR_{B5} \end{bmatrix} \quad (42)$$

7.1 Simulation setup

The objective of the controllers is to maintain the terminal surface temperature error within a 5°C range. However, a 10°C error is considered acceptable for quality control.

The thermoforming model is used as benchmark to test the following three TILC algorithms:

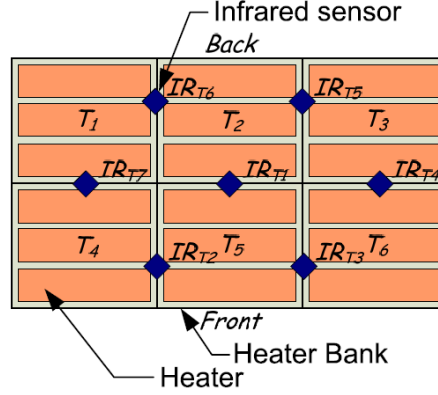


Figure 7: Location of infrared sensors and heaters (bottom sensors and heaters share the same location) [12]

- Crisp 1st order TILC – used as reference for comparison purposes;
- Ideal fuzzy TILC (i-f-TILC) – built from non noisy data;
- Noisy fuzzy TILC (n-f-TILC) – built from noisy data. The noise is assumed to be Gaussian, with variance σ_{noise}^2 on all outputs. For the simulations, the measurement noise of each channel is considered to be independent from that of the others. Since the data contain noise, thirty n-f-TILC were designed and tested to determine the effect of noise on the design.

All the fuzzy TILC were derived from a fuzzy model obtained from the nominal thermoforming model. Three cases were analyzed:

- Case A: target temperature fixed to $\mathbf{y}_d = [160, 150, 150, 160, 150, 150]^T$ °C;
- Case B: target temperature fixed to $\mathbf{y}_d = [140, 140, 140, 140, 140, 140]^T$ °C (uniform surface temperature);
- Case C: target temperature fixed to $\mathbf{y}_d = [160, 150, 150, 160, 150, 150]^T$ °C and σ_{noise} is fixed to 1, 2, 3, 4 and 5 °C.

In cases A and B, the simulations were performed on both nominal and disturbed thermoforming models. The parameters of the two models are shown in Table 1. Both models were tested with and without noise ($\sigma_{noise} = 2^\circ\text{C}$). To enable comparison of the three TILC algorithms, the measurement noise vectors were exactly the same for all simulations. Figure 8 shows the noise added to output y_1 in all simulations with noise. The other noise signals added to the outputs y_2 to y_6 were fairly similar, and were generated with Matlab[®]'s "randn" function.

Table 1: Parameters used in the simulations

Parameters	Units	Nominal value	Disturbed value
Density	kg/m ³	950	1045
Specific heat	J/(kg K)	1838	2022
Effective emissivity		0.45	0.495
Absorptivity	m ⁻¹	300	350
Heat conduction	W/(m K)	0.4	0.3
Convection factor	W/(m ² K)	6	10

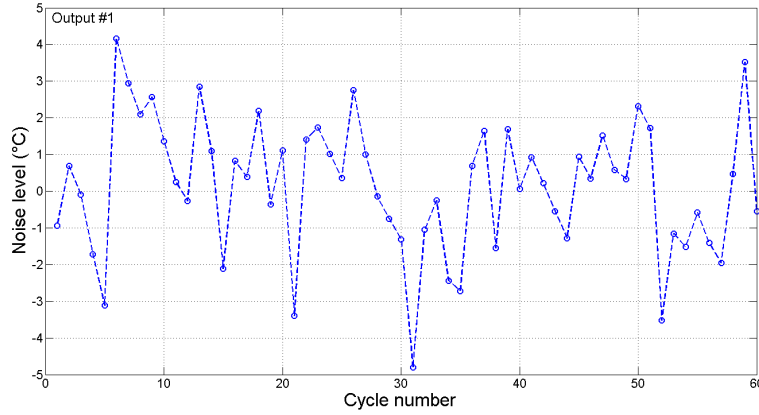


Figure 8: Gaussian noise added to output y_1 (°C) (when $\sigma_{noise} = 2^\circ\text{C}$).

We considered the ambient temperature (in the oven) to be constant in the simulations: $T_{amb} = 125^\circ\text{C}$, or to be subject to drift during the day, so that $T_{amb} = 125^\circ\text{C} + 20^\circ\text{C} \sin(0.0175k)$ (where k is the cycle number). The drift reflects a situation occurring in the industry that forces the operator of the thermoforming machine to retune the heater setpoints to maintain the quality of the parts produced.

Case C is included to determine the effect of different noise levels on the n-f-TILC. Since the fuzzy controller design is affected by noise, we need to check whether or not the noise level has an adverse effect on the design. The results are presented for the worst case, i.e. the disturbed thermoforming model with both noise σ_{noise} and drift. The n-f-TILC has been designed with the same level of noise as used in the simulation. The Kruskal-Wallis statistical test [29] is used to check, with a confidence level of 95% ($\alpha = 0.05$), the similarity of the results. In this test, the \mathcal{H}_0 hypothesis is that all fuzzy TILCs behave similarly from cycle 10 to 60, and the \mathcal{H}_1 hypothesis is that at least one fuzzy TILC behaves differently from the others. The threshold level is $\chi^2_{0.05;30} = 43.77$ for the 31 fuzzy TILCs.

The simulations were performed over 60 cycles ($k \in \{1, \dots, 60\}$) and each

cycle corresponded to 12 mm plastic sheets being heated for 300 seconds (5 minutes).

7.2 Parameters for obtaining the inverse fuzzy model

The fuzzy models of the thermoforming machine were developed by defining a universe of discourse for each input, where $u_{min,k} = 300^\circ\text{C}$ and $u_{max,k} = 450^\circ\text{C}$ (with $k \in \{1, \dots, 6\}$). Each universe of discourse is divided among three fuzzy sets with the peaks located at: $a_{1,k} = 300^\circ\text{C}$, $a_{2,k} = 375^\circ\text{C}$ and $a_{3,k} = 450^\circ\text{C}$ (with $k \in \{1, \dots, 6\}$) respectively.

There are $3^6 = 729$ rules for obtaining each output. This requires $4^6 = 4096$ experiments (simulations, here) to be performed on the thermoforming machine model.

The inverse of the fuzzy model involves evaluating the universe of discourse of all the outputs. The lower limits of those universes of discourse are obtained when all six heating elements are at the lowest setpoint: 300°C . Without measurement noise (and nominal parameters), those lower limits are: $y_{min,k} = 105.06^\circ\text{C}$ (for $k = 1, 3, 4, 6$) and $y_{min,k} = 117.03^\circ\text{C}$ (for $k = 2, 5$). The higher limits are obtained when all the heating elements are at the highest setpoint: 450°C . Again, with no measurement noise, these limits are: $y_{max,k} = 203.89^\circ\text{C}$ (for $k = 1, 3, 4, 6$) and $y_{max,k} = 233.30^\circ\text{C}$ (for $k = 2, 5$). So, as assumed, the minimum and maximum output temperatures are at opposite sides of the input space \mathcal{U} . Since the inputs of the fuzzy model were fuzzified with three fuzzy sets each, the same applies to the inputs of the inverse fuzzy model.

All the fuzzy filters of the fuzzy TILC have normalization gains $K_{N,i} = 0.25$ and denormalization gains $K_{D,i} = 1.00$ for $i \in \{1, \dots, 6\}$.

The two fuzzy TILCs are compared with a so-called crisp TILC based on a linear IMC approach, with an exponential filter [12][28] where the parameters are $\alpha_i = 0.2701 \forall i \in \{1, \dots, 6\}$.

7.3 Simulation results

The simulation results are presented below to compare the behavior of the three TILCs in different situations.

7.3.1 Case A

The first set of simulations has a target temperature vector $\mathbf{y}_d = [160, 150, 150, 160, 150, 150]^T$ $^\circ\text{C}$. The heater temperature setpoints used for the crisp TILC at cycle $k = 1$ are arbitrarily fixed to 350°C for all the heaters. This value is a bad guess, as shown by the surface temperature error at cycle $k = 1$ (or $\|e[1]\|_\infty$) displayed in last column of Tables 2, 3, 4 and 5. It is to be hoped that the crisp TILC algorithm will reduce this error to a lower level as the cycle number k increases. In the ideal case (no noise and no drift), the error converges to zero.

For the two fuzzy TILCs, the initial heater temperature setpoints are obtained from the inverse fuzzy model. With the i-f-TILC, the first heater tem-

perature setpoints are in the vector $\mathbf{sp} = [399.97, 353.22, 403.16, 399.97, 353.22, 403.16]^T$ °C. As shown in Tables 2 to 5, the surface temperature error obtained in the first cycle or $\|e[1]\|_\infty$ is lower than the error obtained with the crisp TILC. Since 30 n-f-TILCs were tested, the values displayed are the mean of the 30 n-f-TILC guesses. The differences between the initial setpoint evaluated by the i-f-TILCs and those evaluated by the 30 n-f-TILCs seem to be negligible.

TILC algorithms must converge to a setpoint vector \mathbf{sp} , such that the surface temperature error $\|e[k]\|_\infty$ falls to 0. When there is no noise, all TILC errors converge to 0. But, when there is noise, there is an error (see μ_e and σ_e in Tables 3 to 5). μ_e is the mean error during cycles 10 to 60, and σ_e is the standard deviation associated with this mean.

Table 2: Simulation results (Case A) — without noise and drift

TILC algorithm	System	$\ e[1]\ _\infty$ (°C)
Crisp TILC	Disturbed	20.7607
	Nominal	18.8756
i-f-TILC	Disturbed	5.5493
	Nominal	1.0671
n-f-TILC (mean of 30 controllers)	Disturbed	5.8967
	Nominal	1.5671

As shown in Table 3, when there is noise ($\sigma_{noise} = 2^\circ\text{C}$), both fuzzy TILCs outperform the crisp TILC by nearly 1.25°C , and the standard deviation σ_e of the surface temperature error is divided by 1.5.

Table 3: Simulation results (Case A) — with noise (μ_e and σ_e for $k \in \{10, \dots, 60\}$)

TILC algorithm	System	μ_e (°C)	σ_e (°C)	$\ e[1]\ _\infty$ (°C)
Crisp TILC	Disturbed	5.1039	1.6734	26.6153
	Nominal	4.9209	1.5830	24.7302
i-f-TILC	Disturbed	3.8572	1.0799	10.5062
	Nominal	3.6603	1.0521	6.9217
n-f-TILC (mean of 30 controllers)	Disturbed	3.8653	1.0887	10.4140
	Nominal	3.6643	1.0539	6.8224

When the ambient temperature is subject to drift and no measurement noise, there is a small error (Table 4). This slowly varying error remains low, and the crisp TILC gives a lower steady state error than either of the fuzzy TILCs.

Table 5 shows the results when we have both noise and drift. The results look similar to the results in Table 3. The drift effect seems relatively small compared to the noise effect. Figure 9 shows the worst case, when the system is subjected to noise, drift, and parameter disturbance. This figure shows that once the TILC algorithms have converged, all the surface temperature errors remain under 12°C . Furthermore, the #1 n-f-TILC's error remains below 6°C

Table 4: Simulation results (Case A) — with drift (μ_e and σ_e for $k \in \{10, \dots, 60\}$)

TILC algorithm	System	μ_e ($^{\circ}\text{C}$)	σ_e ($^{\circ}\text{C}$)	$\ e[1]\ _{\infty}$ ($^{\circ}\text{C}$)
Crisp TILC	Disturbed	0.0549	0.0201	20.7607
	Nominal	0.0417	0.0270	18.8756
i-f-TILC	Disturbed	0.3243	0.2531	5.5493
	Nominal	0.1813	0.0202	1.0671
n-f-TILC (mean of 30 controller)	Disturbed	0.3585	0.3194	5.8967
	Nominal	0.1894	0.0212	1.5671

Table 5: Simulation results (Case A) — with noise and drift (μ_e and σ_e for $k \in \{10, \dots, 60\}$)

TILC algorithm	System	μ_e ($^{\circ}\text{C}$)	σ_e ($^{\circ}\text{C}$)	$\ e[1]\ _{\infty}$ ($^{\circ}\text{C}$)
Crisp TILC	Disturbed	5.0559	1.6482	26.6153
	Nominal	4.9005	1.5933	24.7302
i-f-TILC	Disturbed	3.6883	0.9990	10.5150
	Nominal	3.5901	1.0077	6.9217
n-f-TILC (mean of 30 controllers)	Disturbed	3.6950	1.0122	10.4140
	Nominal	3.5938	1.0059	6.8224

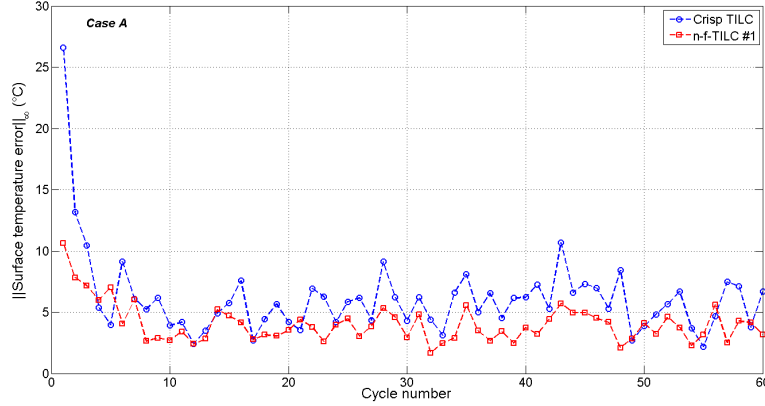


Figure 9: Infinity norm of the surface temperature error in the case of a disturbed system with noise ($\sigma_{noise} = 2^{\circ}\text{C}$) and drift.

after cycle 7.

For each situation covered by Tables 2 to 5, a statistical analysis was performed. The Kruskal-Wallis test indicates that all the fuzzy TILCs give similar results, since the hypothesis \mathcal{H}_0 is not rejected as the worst χ^2 is 0.4780, which is much lower than the threshold $\chi^2_{0.05;30} = 43.77$. Figure 10 shows the boxplots

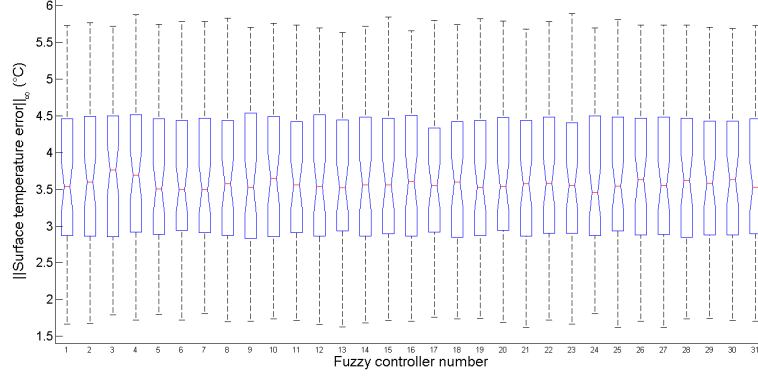


Figure 10: Boxplots of the infinity norm of the temperature error in the case of a disturbed system with noise ($\sigma_{noise} = 2^\circ\text{C}$) and drift. For the 30 n-f-TILCz (#1 to #30) and the i-f-TILC (#31). For cycles 10 to 60.

of the 30 n-f-TILCs and the i-f-TILC (identified as #31) corresponding to the situation covered by Table 5 with the disturbed model. The similarity of the results is obvious when we look at all the boxplots.

7.3.2 Case B

A second set of simulations was performed with a new desired surface temperature vector $\mathbf{y}_d = [140, 140, 140, 140, 140, 140]^T$ °C. When we analyse the results in Table 6 for the situation with noise and drift, we see that the surface temperature errors are in the same order as those obtained in Case A (Table 5). Again, the Kruskal-Wallis test on all the fuzzy TILC results covered by Tables 5 and 6 indicate that all the results obtained with the fuzzy TILCs are similar (with $\chi^2 = 1.0472$). Figure 11 shows the surface temperature error evolution over 60 cycles.

Table 6: Simulation results (Case B) — with noise and drift (μ_e and σ_e for $k \in \{10, \dots, 60\}$)

TILC algorithm	System	μ_e (°C)	σ_e (°C)	$\ e[1]\ _\infty$ (°C)
Crisp TILC	Disturbed	5.5203	1.9141	16.6153
	Nominal	5.3260	1.6303	14.7302
i-f-TILC	Disturbed	3.6363	1.0018	10.5150
	Nominal	3.6148	0.9835	6.9217
n-f-TILC (mean of 30 controllers)	Disturbed	3.6457	1.0204	10.4140
	Nominal	3.6301	0.9949	6.8224

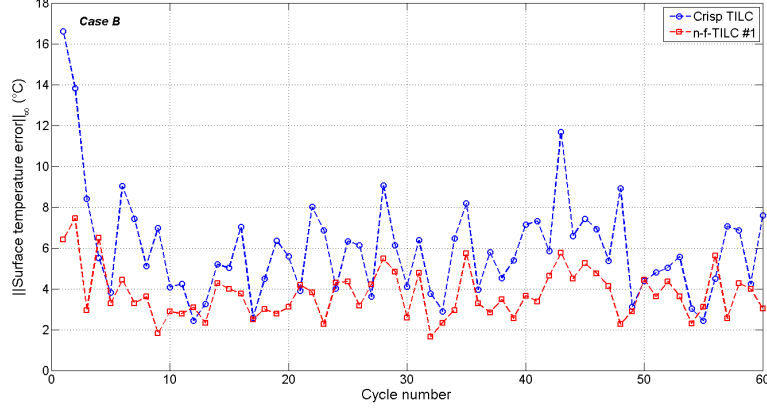


Figure 11: Infinity norm of the surface temperature error in the case of a disturbed system with noise ($\sigma_{noise} = 2^\circ\text{C}$) and drift.

The results show that the desired surface temperature seems to have no significant effect on the error level after cycle 10. Again, the heater setpoints for the crisp TILC at cycle $k = 1$ are arbitrarily fixed to 350°C . Compared to Case A, the initial error is reduced for the crisp TILC, since the desired surface temperature is lower than those requested in Case A.

Table 7: Simulation results — with drift and different levels of noise (All values in $^\circ\text{C}$)

	i-f-TILC		n-f-TILC (σ_{noise})		"Crisp" TILC	
σ_{noise}	μ_e	σ_e	μ_e	σ_e	μ_e	σ_e
1	1.8116	0.5131	1.8117	0.5130	2.6004	0.7985
2	3.5901	1.0077	3.5938	1.0122	5.2955	1.5780
3	5.3732	1.5737	5.3945	1.5689	8.1212	2.4245
4	7.1541	2.1611	8.6887	2.8256	11.0789	3.4452
5	8.9642	2.7093	9.8786	3.2403	14.2144	4.6113
5*	-	-	9.5391	3.0152	-	-

7.3.3 Case C

In the last set of simulations, the three TILCs were tested for different noise levels ($\sigma_{noise} \in \{1, 2, 3, 4, 5\}^\circ\text{C}$). The results are displayed in Table 7 when drift is present, and in Table 8 Table VIII when there is no drift. In all the simulations, the crisp TILC has a higher temperature error than the two fuzzy TILCs. Apparently, both fuzzy TILCs perform with approximately the same error level.

Table 8: Simulation results — with different levels of noise (All values in °C and no drift)

	i-f-TILC		n-f-TILC (σ_{noise})		"Crisp" TILC	
σ_{noise}	μ_e	σ_e	μ_e	σ_e	μ_e	σ_e
1	1.8405	0.5083	1.8408	0.5085	2.6227	0.8006
2	3.6603	1.0521	3.6643	1.0539	5.3443	1.5825
3	5.4803	1.6581	5.5022	1.6523	8.2013	2.4334
4	7.3448	2.2510	8.6489	2.9075	11.1976	3.4583
5	9.1766	2.8496	10.0518	3.3427	14.3811	4.6158
5*	-	-	9.2790	2.9458	-	-

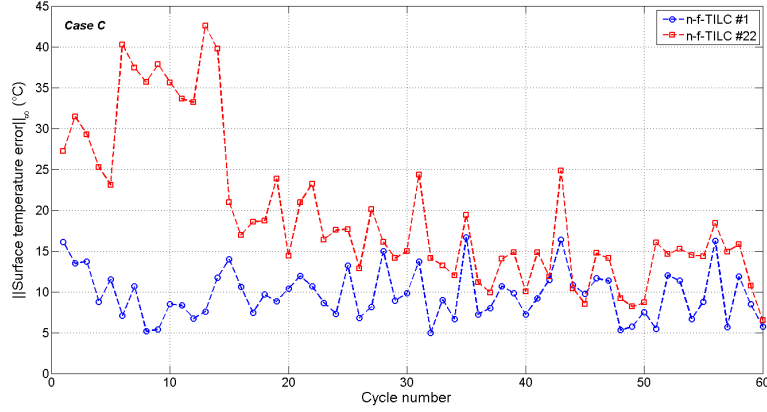


Figure 12: Infinity norm of the surface temperature error in the case of a disturbed system with noise ($\sigma_{noise} = 5$).

Again using the Kruskal-Wallis test on all the fuzzy TILC results in Tables 7 and 8, we find that we can't reject hypothesis \mathcal{H}_0 (χ_2 goes from 0.2614 to 1.3564 and is much lower than $\chi_{0.05;30}^2$), except when $\sigma_{noise} = 5^\circ\text{C}$, where hypothesis \mathcal{H}_0 is rejected (χ_2 goes from 73.5555 to 90.7538 and is higher than $\chi_{0.05;30}^2$). For the noisy case (with no drift), there are two outliers in the n-f-TILC, and, for the case with noise and drift, there is one outlier. By removing these outliers, we reduce the error in the case where $\sigma_{noise} = 5^\circ\text{C}$, since the remaining results are similar. This corresponds to lines where σ_{noise} is identified as 5* in Tables 7 and 8. Figure 12 shows the results for two n-f-TILC algorithms when $\sigma_{noise} = 5^\circ\text{C}$. n-f-TILC #22 is an outlier, while n-f-TILC #1 is similar to the majority of n-f-TILCs (see the boxplots in Figure 13).

It is hoped that the noise level (σ_{noise}) will be somewhere between 1 and 2°C on the real process. The simulation results show that, if the measurement signals are very noisy, more than one experiment must be performed for each input tuple, and use the mean value of the output measurements to build the

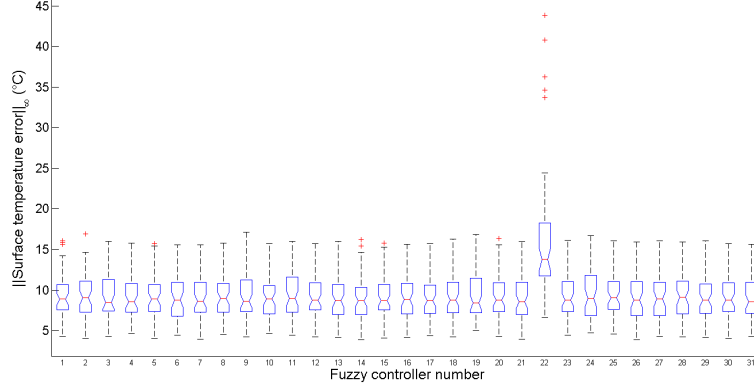


Figure 13: Boxplots of the infinity norm of the temperature error in the case of a disturbed system with noise ($\sigma_{noise} = 5^\circ\text{C}$) and drift. For the 30 n-f-TILCz (#1 to #30) and the i-f-TILC (#31). For cycles 10 to 60.

fuzzy model.

The simulation results also indicate that every measurement signal must be filtered. When the noise level is low, the n-f-TILC seems to be relatively robust to this noise, since the resulting n-f-TILC algorithms behave in a similar way to the i-f-TILC algorithm.

If the 10°C target for the error after cycle 10 is easy to obtain with a fuzzy TILC when the noise level σ_{noise} is below 3°C , the 5°C target is obtained only for σ_{noise} equal to 1°C . This means that it is necessary to filter the measurement signals. The ambient temperature seems to have relatively little effect on the convergence of TILC algorithms in the presence of noise.

8 Conclusion

This paper has shown the good results obtained using a fuzzy TILC when compared to a 1st order crisp TILC, even in the case where the data used to build the fuzzy TILC were contaminated with a low level of noise.

The proposed design approach provides an easier way to obtain an inverse fuzzy model than the TSK of order 0 for systems with a large number of inputs and outputs, since the fuzzy model relies on linear rules expressed by matrices when all the outputs are combined into a vector.

The main problem with fuzzy modeling (with a TSK of any order) is the huge number of experiments needed to obtain the data necessary to build the fuzzy model. In a real process, this translates to a significant waste of plastic. With simulation, the only cost is calculation time.

Since thermoforming machines are used for the production of plastic parts,

the data can be obtained from measurements performed on the actual production process. Building a fuzzy model from those measurements is work that we will need to do in the future.

Another potential problem is the effect of noise when the data used to build fuzzy model are obtained from measurements performed on a real process. Optimistically, for the real thermoforming process, the noise level approximately corresponds to the simulations where $\sigma_{noise} = 1^\circ\text{C}$. However, noise can become an issue for other processes. Research is needed to analyze how noise can be reduced. A possible solution to this problem is to perform more than one measurement for each input tuple and to use the mean of those measurements as data.

In future work, it will be interesting to analyze how the fuzzy model is obtained and inverted for the case where the number of inputs is not equal to the number of outputs. Moreover, since the simulation results are quite promising, we want to test the fuzzy TILC on an industrial thermoforming machine, as we are convinced that this control can be used safely, without damaging the machine (by overheating).

9 Appendix [30]

This appendix is taken from [30].

The nonlinear model of the thermoforming oven used in the simulations in section VII has been reproduced from the Ph.D. thesis of one of the authors [12]. In the model of the thermoforming oven, the plastic sheet is divided into zones, one for each sensor pair (Figure 14).

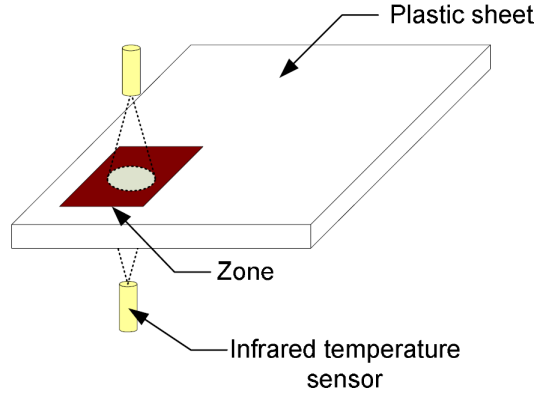


Figure 14: IR temperature sensors and the corresponding zone ([12])

To analyze the temperature behavior throughout the plastic sheet, each zone is divided into five layers, and each layer has a node (Figure 15). Lateral conduction between nodes of adjacent zones is neglected in this model.

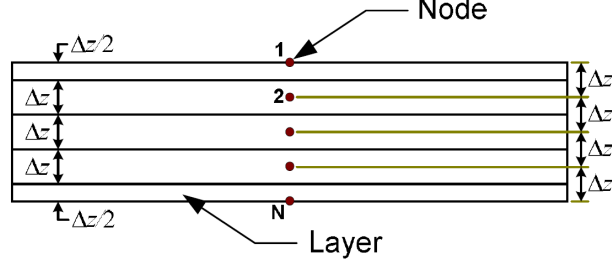


Figure 15: Layers and nodes ([12])

We present the mathematical model first, and then we explain its parameters.

For node 1, in zone k at the top surface of the plastic sheet, the temperature dynamic is expressed by:

$$\frac{dT_{k,1}}{dt} = \frac{2}{\rho V C_p} \left\{ \beta_1 Q_{RT_k} + \beta_1 (1 - \beta_1) (1 - \beta_2)^3 Q_{RB_k} \right. \\ \left. + h \{T_\infty - T_{k,1}\} + \frac{kA}{\Delta z} \{T_{k,2} - T_{k,1}\} \right\} \quad (43)$$

Similarly, for node 5, at the bottom surface of the plastic sheet:

$$\frac{dT_{k,5}}{dt} = \frac{2}{\rho V C_p} \left\{ \beta_1 (1 - \beta_1) (1 - \beta_2)^3 Q_{RT_k} + \beta_1 Q_{RB_k} \right. \\ \left. + h \{T_\infty - T_{k,5}\} + \frac{kA}{\Delta z} \{T_{k,4} - T_{k,5}\} \right\} \quad (44)$$

For the internal nodes i (of zone k), located inside the plastic sheet, we have the following dynamic:

$$\frac{dT_{k,i}}{dt} = \frac{1}{\rho V C_p} \left\{ \beta_2 (1 - \beta_1) \left\{ (1 - \beta_2)^{i-2} Q_{RT_k} + (1 - \beta_2)^{4-i} Q_{RB_k} \right\} \right. \\ \left. + \frac{kA}{\Delta z} \{T_{k,i-1} - 2T_{k,i} + T_{k,i+1}\} \right\} \quad (45)$$

The radiant terms in (43), (44), and (45) are defined for the top heaters as follows:

$$Q_{RT_k} = \sigma \epsilon_{eff} A_h \sum_{j=1}^6 F_{kj} \{\theta_j^4 - T_{k,1}^4\} \quad (46)$$

and for the bottom heaters as follows:

$$Q_{RB_k} = \sigma \epsilon_{eff} A_h \sum_{j=7}^{12} F_{kj} \{\theta_j^4 - T_{k,5}^4\} \quad (47)$$

These are the parameters that appear in the Appendix equations:

- A : zone surface, in m^2 ;
- A_h : heater surface, in m^2 ;
- β_1 : fraction of radiant energy absorbed by a surface layer;

- β_2 : fraction of radiant energy absorbed by an internal layer;
- C_P : specific heat of the plastic sheet, in J/(kg.K);
- ϵ_{eff} : effective emissivity;
- F_{kj} : view factor between heater j and sheet zone k ;
- h : convection coefficient, in W/(m².K);
- k : heat conduction constant, in W/(m.K);
- ρ : density of the plastic sheet, in kg/m³;
- T_∞ : oven ambient temperature, in K;
- $T_{k,i}$: temperature of node i in zone k , in K;
- θ_j : temperature of heater j , in K;
- σ : Stefan Boltzmann constant equal to 5.669×10^{-8} W/(m².K⁴);
- Δz : layer thickness, in meters;
- V : layer volume, in m³; ($V = A\Delta z$);

Additional details about this model can be found in chapter 2 in [12] or chapter 3 in [4].

References

- [1] J. L. Throne and P. J. Mooney, “Thermoforming: Growth and evolution 1.” <http://www.foamandform.com/technical-minutes/thermoforming/thermoforming-growth-and-evolution-i>, November 2002. Accessed: May 20, 2011.
- [2] J. L. Throne, *Technology of thermoforming*. Hanser Publishers, 1996.
- [3] J. Florian, *Practical Thermoforming: Principles and Applications*. Marcel Dekker Inc., 1987.
- [4] M. Ajersch, “Modeling and real-time control of sheet reheat phase in thermoforming,” Master’s thesis, McGill University, 2004.
- [5] M. M. I. Chy and B. Boulet, “A conjugate gradient method for the solution of the inverse heating problem in thermoforming,” in *2010 IEEE Industry Applications Society Annual Meeting*, pp. 1–8, Oct 2010.
- [6] P. Girard, R. di Raddo, V. Thomson, and B. Boulet, “Advanced in-cycle and cycle-to cycle on-line adaptive control for thermoforming of large thermoplastic sheets,” in *SAE Technical Paper*, SAE International, 04 2005.

- [7] F. M. Duarte and J. A. Covas, "Ir sheet heating in roll fed thermoforming: Part 1 - solving direct and inverse heating problems," *Plastics, Rubber and Composites*, vol. 31, no. 7, pp. 307–317, 2002.
- [8] Y. Chen, J.-X. Xu, and C. Wen, "A high-order terminal iterative learning control scheme [rtp-cvd application]," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 4, pp. 3771–3772, Dec 1997.
- [9] Y. Chen, *High-Order Iterative Learning Control: Convergence, Robustness and Applications*. PhD thesis, Nanyang Technological University: Singapore, 1997.
- [10] K. L. Moore, *Iterative Learning Control: An Expository Overview*, pp. 151–214. Boston, MA: Birkhäuser Boston, 1999.
- [11] Y. Chen and C. Wen, eds., *High-order terminal iterative learning control with an application to a rapid thermal process for chemical vapor deposition*, pp. 95–104. London: Springer London, 1999.
- [12] G. Gauthier, *Terminal iterative learning for cycle-to-cycle control of industrial processes*. PhD thesis, Department of Electrical and Computer Engineering. McGill University: Montreal, Canada, 2008.
- [13] C. Zhang, H. Deng, and J. Baras, "Comparison of run-to-run control methods in semiconductor manufacturing processes," in *AEC/APC Symposium XII, Lake Tahoe, Nevada*, 2000.
- [14] J. Moyne, E. del Castillo, and A. Hurwitz, *Run-to-Run Control in Semiconductor Manufacturing*. CRC Press, 2001.
- [15] S. Adivikolanu and E. Zafiriou, "Robust run-to-run control for semiconductor manufacturing: an internal model control approach," in *Proceedings of the 1998 American Control Conference*, vol. 6, pp. 3687–3691, Jun 1998.
- [16] E. Sachs, A. Hu, and A. Ingolfsson, "Run by run process control: combining spc and feedback control," *IEEE Transactions on Semiconductor Manufacturing*, vol. 8, pp. 26–43, Feb 1995.
- [17] F. M. Duarte and J. A. Covas, "Ir sheet heating in roll fed thermoforming: Part 2 – factors influencing inverse heating solution," *Plastics, Rubber and Composites*, vol. 32, no. 1, pp. 32–39, 2003.
- [18] M. M. I. Chy and B. Boulet, "A new method for estimation and control of temperature profile over a sheet in thermoforming process," in *2010 IEEE Industry Applications Society Annual Meeting*, pp. 1–8, Oct 2010.
- [19] M. Chy, B. Boulet, and G. Gauthier, *Estimation and control of sheet temperature in thermoforming*, ch. 9. Optimization of Polymer Processing, Nova Science Publishers, Inc, 2011.

- [20] R. Boukezzoula, S. Galichet, and L. Foulloy, "Nonlinear internal model control: application of inverse model based fuzzy control," *IEEE Transactions on Fuzzy Systems*, vol. 11, pp. 814–829, Dec 2003.
- [21] R. Boukezzoula, S. Galichet, and L. Foulloy, "Exact inversion of takagi-sugeno fuzzy models," in *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, vol. 4, pp. 2108–2113, July 2001.
- [22] S. A. Salman and G. S. Anavatti, "A novel inversion method for a fuzzy model based controller," in *2007 International Conference on Mechatronics and Automation*, pp. 100–104, Aug 2007.
- [23] S. Galichet, R. Boukezzoula, and L. Foulloy, "Explicit analytical formulation and exact inversion of decomposable fuzzy systems with singleton consequents," *Fuzzy Sets and Systems*, vol. 146, no. 3, pp. 421 – 436, 2004.
- [24] M. Fu, Z. Ding, G. Xia, and K. Fan, "Research on the fuzzy decoupling in the coordinated control system of nuclear power plant," in *2008 IEEE International Conference on Automation and Logistics*, pp. 405–410, Sept 2008.
- [25] A. Fortin, *Analyse Numérique pour ingénieurs*. Presses Internationales Polytechnique, Montreal, 2001.
- [26] M. Beauchemin-Turcotte, "Contrôle en fin de cycle par apprentissage itératif via la logique floue appliquée au contrôle en température d'un four de thermoformage," Master's thesis, Department of Automated Manufacturing Engineering, École de Technologie Supérieure, 2010.
- [27] B. Oggunnaike and W. Ray, *Process Dynamics, Modeling, and Control*. Oxford University Press, 1994.
- [28] M. Morari and E. Zafiriou, *Robust Process Control*. Prentice Hall, 1989.
- [29] R. Rakotomalada, "Comparaison de populations – tests non paramétriques." http://eric.univ-lyon2.fr/~ricco/cours/cours/Comp_Pop_Tests_Nonparametriques.pdf, August 2008. Accessed: March 09, 2017.
- [30] G. Gauthier and B. Boulet, "Terminal iterative learning control with internal model control applied to the thermoforming reheat phase," in *AFRICON 2009*, pp. 1–7, Sept 2009.